UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/729,518	12/05/2003	Guy Herriott	200314936-1	2391

22879 7590 04/17/2008
HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
SMITH, JOSHUA Y	

ART UNIT	PAPER NUMBER
2619	

NOTIFICATION DATE	DELIVERY MODE
04/17/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM
mkraft@hp.com
ipa.mail@hp.com

Office Action Summary	**Application No.** 10/729,518	**Applicant(s)** HERRIOTT ET AL.
	Examiner JOSHUA SMITH	**Art Unit** 2619

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *30 January 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2 and 4-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2 and 4-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 01/30/2008 has been entered.

- **Claims 1, 2 and 4-20 are pending.**
- **Claim 3 was previously cancelled.**
- **Claims 1, 2 and 4-20 stand rejected.**

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

    1. Determining the scope and contents of the prior art.

2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

3. **Claims 1, 5, 7-10 and 16** are rejected under 35 U.S.C. 103(a) as being unpatentable over Park (Pub. No.: US 2002/0085554 A1) in view of Navada et al. (Pub. No.: US 2003/0214956 A1), Musoll et al. (Patent No.: US 7,155,516 B2), Ambe et al. (Pub. No.: US 2002/0196796 A1), and Zhang (Patent No.: US 6,845,094 B1), hereafter referred to as Park, Navada, Musoll, Ambe, and Zhang, respectively.

As for Claim 1, Park teaches in paragraph [0034] of a "routing device 20 receives a packet from another node or routing device" (substantively the same as "receiving the data packet at a network device" in the instant invention).

Park shows in paragraph [0060] and in FIG. 5, Sheet 3 of 3, "if it is determined from the step S10 that the destination address of the packet is identical to the cache address found in the step S9" (substantively the same as "determining whether a … cache is hit by the data packet" in the instant invention), then "the main processor 11 sends the packet to the interface corresponding to the cache address" (substantively the same as "applying at least one cached action if the decision cache is hit" in the instant invention).

Park shows in paragraph [0056] and in FIG. 5, Sheet 3 of 3, "If the destination address of the packet is not identical to the cache IP address found in the step S9, the main processor 11 concludes that the destination address does not exist in the main cache table 11A. Therefore, it sends the packet to the protocol layer 12A (S30)", which "may be any one of the IP (Internet Protocol) layer 2A, IPX (Internetwork Packet

exchange) layer 2B, Bridge layer 2C, and many others" (see paragraph [0006] and FIG. 1, Sheet 1 of 3, of Park) (substantively the same as "processing the data packet using software routines if the decision cache is missed" in the instant invention).

Park does not teach a networking switch, "multiple-key", determining whether an action by software is programmable into cache, and a new entry indexes into a cache, and programming a new entry does not involve storing a data packet, and a cached action comprises network address translation of a data packet. Navada teaches "multiple-key" and a new entry indexes into a cache and a networking switch, Musoll teaches determining whether an action by software is programmable into a cache, Ambe teaches programming a new entry does not involve storing a data packet, and Zhang teaches a cached action comprises network address translation of a data packet.

In the same field of endeavor, Navada teaches in paragraphs [0028] and [0002], and Fig. 2, Sheet 2 of 8, "the key 216 may be a hardware address, software address, and/or VLAN tag", and "keys are typically Internet protocol (IP) addresses, media access control (MAC) addresses, virtual local area network (VLAN) tags, and other network identifiers" (substantively the same as "multiple-key" in the instant invention).

Navada also teaches in paragraph [0028], "content that is obtained from the key 216 is used to directly address and/or index into a table/memory location" (substantively the same as "new entry indexes into a decision cache" in the instant invention).

Navada also teaches in paragraphs [0018] and [0019], networking devices that store and retrieve data and contain translation tables, and a network that includes a router and switches (the same as "a networking switch" in the instant invention). It

would have been obvious to one skilled in the art at the time of the invention to adopt the method of Navada into the method of Park since the method of Park does not explicitly involve fast VLAN lookups, while the method of Navada explicitly provides a detailed method of memory efficient fast VLAN lookups.

In the same field of endeavor, Musoll teaches in lines 36-40, column 4, “steps of (a) attempting to store all incoming packets, by a first storage system, into a local packet memory (LPM)” and “(b) relinquishing packets incompatible with the LPM to a second storage system” (substantively the same as “determining whether action performed by the software routines is programmable into the multiple-key decision cache” and “programming a new entry into the multiple-key decision cache if the action performed is programmable” in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to combine the method of Musoll with the method of Park since, if the method of Park is implemented in hardware, Musoll provides a method of taking packets that cannot be processed by the hardware and efficiently conveying them to software-based processes.

In the same field of endeavor, Ambe teaches in paragraph [0167], an FFP is programmed by a user, where a user identifies the protocol fields of a packet which are to be of interest for a filter, and after a filter mask is constructed, it is determined whether the filter will be an inclusive or exclusive filter, depending upon the problems which are sought to be solved, the packets which are sought to be forwarded, actions sought to be taken, ect., and based on the steps in FIG. 17, Sheet 17 of 20, a rules table entry for a rules table is then constructed (substantively the same as

“programming the new entry does not involve storing the data packet” in the instant invention). It would have been obvious to one of ordinary skill in the art to combine the invention of Ambe with the invention of Park since Ambe provides a filter that can be programmed for use in routing packets and where the rules and information programmed in are stored in a rules table entry, saving the rules and information for future use in treating and forwarding certain packets or packet types, expanding the routing capabilities of Park.

In the same field of endeavor, Zhang teaches in column 6, lines 7-17, and in column 7, lines 6-13 and 18-26, network address translation between a dynamic IP address space and a permanent IP address space using stored network address translation (NAT) information, and where NAT information comprises a cache of mobility information generally including a plurality of cache entries, and where NAT information is stored and managed in a distributed cache (substantively the same as “a cached action comprises network address translation of a data packet” in the instant invention). It would have been obvious to one of ordinary skill in the art to combine the invention of Zhang with the invention of Park since Zhang provides a system of a cache that contains network address translation information, which can be incorporated into the system of Park to allow the cache of Part to accommodate NAT so that the system of Park can support NAT when it is implemented as a node between disparate networks that have different addressing types or where one network is implementing private addressing.

As for Claim 5, Park does not teach generating a hash value from multiple fields in the data packet, and using the hash value generated to index into the multiple-key decision cache. Navada teaches these limitations. Navada teaches in paragraph [0028], "the key 216 may be a hardware address, software address, and/or VLAN tag included in the header of a data packet/datagram" and that "The content that is obtained from the key 216 is used to directly address and/or index into a table/memory location". Navada also teaches in paragraph [0029], "the reader 206 may also obtain the content by hashing all or part of the key" (substantively the same as "generating a hash value from multiple fields in the data packet" and "using the hash value generated to index into the multiple-key decision cache" in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt the method of Navada into the method of Park since the method of Park does not explicitly involve fast VLAN lookups, while the method of Navada explicitly provides a detailed method of memory efficient fast VLAN lookups.

As for Claim 7, Park teaches in paragraphs [0053] and [0060], and FIG. 5, Sheet 3 of 3, "main processor 11 finds the cache address corresponding to the calculated Hashing key by searching the main cache table 11A (S9)", and "if it is determined from the step S10 that the destination address of the packet is identical to the cache address found in the step S9, the main processor 11 sends the packet to the interface corresponding to the cache address (S1)" (substantively the same as "if the hash entry is valid in the ... decision cache, then determining whether pertinent fields of

the data packet exactly match corresponding fields of the entry" and "if the pertinent fields exactly match, then providing a result that the decision cache is hit" in the instant invention). Park fails to teach "multiple-key". Navada teaches "multiple-key".

In the same field of endeavor, Navada teaches in paragraphs [0028] and [0002], and Fig. 2, Sheet 2 of 8, "the key 216 may be a hardware address, software address, and/or VLAN tag", and "keys are typically Internet protocol (IP) addresses, media access control (MAC) addresses, virtual local area network (VLAN) tags, and other network identifiers" (substantively the same as "multiple-key" in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt the method of Navada into the method of Park since the method of Park does not explicitly involve fast VLAN lookups, while the method of Navada explicitly provides a detailed method of memory efficient fast VLAN lookups.

As for Claim 8, as discussed above with respect to Claim 1, Park teaches a method. Park fails to teach search keys for the decision cache include source and destination IP addresses. Navada teaches these limitations. Navada teaches in paragraph [0002], "keys are typically Internet protocol (IP) addresses" (substantively the same as "search keys for the decision cache include source and destination IP addresses" in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt the method of Navada into the method of Park since the method of Park does not explicitly involve fast VLAN lookups, while the method of Navada explicitly provides a detailed method of memory efficient fast VLAN lookups.

As for Claim 9, as discussed above with respect to Claim 1, Park teaches a method. Park fails to teach search keys further include an inbound VLAN identifier. Navada teaches these limitations. Navada teaches in paragraph [0028], "the key 216 may be a ... VLAN tag" (substantively the same as "search keys further include an inbound VLAN identifier" in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt the method of Navada into the method of Park since the method of Park does not explicitly involve fast VLAN lookups, while the method of Navada explicitly provides a detailed method of memory efficient fast VLAN lookups.

As for Claim 10, as discussed above with respect to Claim 1, Park teaches a method. Park fails to teach search keys for the decision cache include source MAC addresses. Navada teaches these limitations. Navada teaches in paragraph [0002], "keys are typically ... media access control (MAC) addresses" (substantively the same as "search keys for the decision cache include source MAC addresses" in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt the method of Navada into the method of Park since the method of Park does not explicitly involve fast VLAN lookups, while the method of Navada explicitly provides a detailed method of memory efficient fast VLAN lookups.

As for Claim 16, Park teaches in paragraph [0034] of a "routing device 20 receives a packet from another node or routing device" (substantively the same as "receiving the data packet at a network device" in the instant invention).

Park shows in paragraph [0060] and in FIG. 5, Sheet 3 of 3, "if it is determined from the step S10 that the destination address of the packet is identical to the cache address found in the step S9" (substantively the same as "determining whether a ... cache is hit by the data packet" in the instant invention), then "the main processor 11 sends the packet to the interface corresponding to the cache address" (substantively the same as "applying at least one cached action if the decision cache is hit" in the instant invention).

Park shows in paragraph [0056] and in FIG. 5, Sheet 3 of 3, "If the destination address of the packet is not identical to the cache IP address found in the step S9, the main processor 11 concludes that the destination address does not exist in the main cache table 11A. Therefore, it sends the packet to the protocol layer 12A (S30)", which "may be any one of the IP (Internet Protocol) layer 2A, IPX (Internetwork Packet exchange) layer 2B, Bridge layer 2C, and many others" (see paragraph [0006] and FIG. 1, Sheet 1 of 3, of Park) (substantively the same as "processing the data packet using software routines if the decision cache is missed" in the instant invention).

Park further shows in paragraph [0006], that every item 6 in FIG. 1, Sheet 1 of 3, that "a packet is received through a port 6", and port item 6 is a port among multiple ports (substantively the same as "a plurality of ports configured to receive data packets" in the instant invention).

Park does not teach a networking switch, “multiple-key”, determining whether an action by software is programmable into cache, and a new entry indexes into a cache, and programming a new entry does not involve storing a data packet, and a cached action comprises network address translation of a data packet. Navada teaches “multiple-key” and a new entry indexes into a cache and a networking switch, Musoll teaches determining whether an action by software is programmable into a cache, and Ambe teaches programming a new entry does not involve storing a data packet, and Zhang teaches a cached action comprises network address translation of a data packet.

In the same field of endeavor, Navada teaches in paragraphs [0028] and [0002], and Fig. 2, Sheet 2 of 8, “the key 216 may be a hardware address, software address, and/or VLAN tag”, and “keys are typically Internet protocol (IP) addresses, media access control (MAC) addresses, virtual local area network (VLAN) tags, and other network identifiers” (substantively the same as “multiple-key” in the instant invention).

Navada also teaches in paragraph [0028], “content that is obtained from the key 216 is used to directly address and/or index into a table/memory location” (substantively the same as “new entry indexes into a decision cache” in the instant invention).

Navada further teaches in paragraph [0050], of a packet switch that comprises “routines, subroutines, components, subcomponents, registers, processors, circuits, software subroutines, and/or software objects, or any combination thereof” (substantively the same as “software routines configures to process the data packets” in the instant invention).

Navada also teaches in paragraphs [0018] and [0019], networking devices that store and retrieve data and contain translation tables, and a network that includes a router and switches (the same as “a networking switch” in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt the method of Navada into the method of Park since the method of Park does not explicitly involve fast VLAN lookups, while the method of Navada explicitly provides a detailed method of memory efficient fast VLAN lookups.

In the same field of endeavor, Musoll teaches in lines 36-40, column 4, “steps of (a) attempting to store all incoming packets, by a first storage system, into a local packet memory (LPM)” and “(b) relinquishing packets incompatible with the LPM to a second storage system” (substantively the same as “determining whether action performed by the software routines is programmable into the multiple-key decision cache” and “programming a new entry into the multiple-key decision cache if the action performed is programmable” in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to combine the method of Musoll with the method of Park since, if the method of Park is implemented in hardware, Musoll provides a method of taking packets that cannot be processed by the hardware and efficiently conveying them to software-based processes.

In the same field of endeavor, Ambe teaches in paragraph [0167], an FFP is programmed by a user, where a user identifies the protocol fields of a packet which are to be of interest for a filter, and after a filter mask is constructed, it is determined whether the filter will be an inclusive or exclusive filter, depending upon the problems

which are sought to be solved, the packets which are sought to be forwarded, actions sought to be taken, ect., and based on the steps in FIG. 17, Sheet 17 of 20, a rules table entry for a rules table is then constructed (substantively the same as "programming the new entry does not involve storing the data packet" in the instant invention). It would have been obvious to one of ordinary skill in the art to combine the invention of Ambe with the invention of Park since Ambe provides a filter that can be programmed for use in routing packets and where the rules and information programmed in are stored in a rules table entry, saving the rules and information for future use in treating and forwarding certain packets or packet types, expanding the routing capabilities of Park.

In the same field of endeavor, Zhang teaches in column 6, lines 7-17, and in column 7, lines 6-13 and 18-26, network address translation between a dynamic IP address space and a permanent IP address space using stored network address translation (NAT) information, and where NAT information comprises a cache of mobility information generally including a plurality of cache entries, and where NAT information is stored and managed in a distributed cache (substantively the same as "a cached action comprises network address translation of a data packet" in the instant invention). It would have been obvious to one of ordinary skill in the art to combine the invention of Zhang with the invention of Park since Zhang provides a system of a cache that contains network address translation information, which can be incorporated into the system of Park to allow the cache of Part to accommodate NAT so that the system of Park can support NAT when it is implemented as a node between disparate networks

that have different addressing types or where one network is implementing private addressing.

Claims 2, 17 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Park in view of Navada, Musoll, Ambe, Zhang, and further in view of Kadambi et al. (Patent No.: US 6,643,261 B2), hereafter referred to as Kadambi.

As for Claim 2, Park does not teach determining capability of hardware circuitry. Kadambi teaches these limitations. In the same field of endeavor, Kadambi teaches in lines 23-24, 29-30, 44-47, column 27, of a device that receives a packet and "If the packet is identified as neither an IP packet nor an IPX packet, the packet is directed to CPU", but the device could otherwise process a received IP packet or IPX packet since it can provide "support of both IP and IPX protocol", where this "capability is provided within logic circuitry". Kadambi teaches in lines 35-38, column 27, that this identification of packet type is done before cache lookup since packet type influences lookups (substantively the same as "prior to determining whether the multiple-key decision is hit" and "determining whether hardware circuitry of the network device is capable of processing the data packet" and "processing the data packet using hardware circuitry if the hardware circuitry is determined to be capable" in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt a form of the method of Park into the architecture of Kadambi since Kadambi provides a detailed architecture where at least aspects of Parks method can be implemented since both endeavor to improve the rate of forwarding though the use of caches and hashing.

As for Claim 17, Park does not teach determining capability of hardware circuitry. Kadambi teaches these limitations. In the same field of endeavor, Kadambi teaches in lines 23-24, 29-30, 44-47, column 27, of a device that receives a packet and "If the packet is identified as neither an IP packet nor an IPX packet, the packet is directed to CPU", but the device could otherwise process a received IP packet or IPX packet since it can provide "support of both IP and IPX protocol", where this "capability is provided within logic circuitry". Kadambi teaches in lines 35-38, column 27, that this identification of packet type is done before cache lookup since packet type influences lookups (substantively the same as "prior to determining whether the multiple-key decision is hit" and "determining whether hardware circuitry of the network device is capable of processing the data packet" and "processing the data packet using hardware circuitry if the hardware circuitry is determined to be capable" in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt a form of the method of Park into the architecture of Kadambi since Kadambi provides a detailed architecture where at least aspects of Parks method can be implemented since both endeavor to improve the rate of forwarding though the use of caches and hashing.

As for Claim 18, Park teaches in paragraph [0034] of a "routing device 20 receives a packet from another node or routing device" (substantively the same as "receiving the data packet at a network device" in the instant invention).

Park shows in paragraph [0060] and in FIG. 5, Sheet 3 of 3, "if it is determined from the step S10 that the destination address of the packet is identical to the cache address found in the step S9" (substantively the same as "determining whether a … cache is hit by the data packet" in the instant invention), then "the main processor 11 sends the packet to the interface corresponding to the cache address" (substantively the same as "applying at least one cached action if the decision cache is hit" in the instant invention).

Park shows in paragraph [0056] and in FIG. 5, Sheet 3 of 3, "If the destination address of the packet is not identical to the cache IP address found in the step S9, the main processor 11 concludes that the destination address does not exist in the main cache table 11A. Therefore, it sends the packet to the protocol layer 12A (S30)", which "may be any one of the IP (Internet Protocol) layer 2A, IPX (Internetwork Packet exchange) layer 2B, Bridge layer 2C, and many others" (see paragraph [0006] and FIG. 1, Sheet 1 of 3, of Park) (substantively the same as "processing the data packet using software routines if the decision cache is missed" in the instant invention).

Park does not teach a networking switch, "multiple-key", determining whether an action by software is programmable into cache, and a new entry indexes into a cache, and programming a new entry does not involve storing a data packet, and a cached action comprises network address translation of a data packet. Navada teaches "multiple-key" and a new entry indexes into a cache and a networking switch, Musoll teaches determining whether an action by software is programmable into a cache, and

Ambe teaches programming a new entry does not involve storing a data packet, and Zhang teaches a cached action comprises network address translation of a data packet.

In the same field of endeavor, Navada teaches in paragraphs [0028] and [0002], and Fig. 2, Sheet 2 of 8, “the key 216 may be a hardware address, software address, and/or VLAN tag”, and “keys are typically Internet protocol (IP) addresses, media access control (MAC) addresses, virtual local area network (VLAN) tags, and other network identifiers” (substantively the same as “multiple-key” in the instant invention).

Navada also teaches in paragraph [0028], “content that is obtained from the key 216 is used to directly address and/or index into a table/memory location” (substantively the same as “new entry indexes into a decision cache” in the instant invention).

Navada also teaches in paragraphs [0018] and [0019], networking devices that store and retrieve data and contain translation tables, and a network that includes a router and switches (the same as “a networking switch” in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt the method of Navada into the method of Park since the method of Park does not explicitly involve fast VLAN lookups, while the method of Navada explicitly provides a detailed method of memory efficient fast VLAN lookups.

In the same field of endeavor, Musoll teaches in lines 36-40, column 4, “steps of (a) attempting to store all incoming packets, by a first storage system, into a local packet memory (LPM)” and “(b) relinquishing packets incompatible with the LPM to a second storage system” (substantively the same as “determining whether action performed by the software routines is programmable into the multiple-key decision

cache" and "programming a new entry into the multiple-key decision cache if the action performed is programmable" in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to combine the method of Musoll with the method of Park since, if the method of Park is implemented in hardware, Musoll provides a method of taking packets that cannot be processed by the hardware and efficiently conveying them to software-based processes.

In the same field of endeavor, Ambe teaches in paragraph [0167], an FFP is programmed by a user, where a user identifies the protocol fields of a packet which are to be of interest for a filter, and after a filter mask is constructed, it is determined whether the filter will be an inclusive or exclusive filter, depending upon the problems which are sought to be solved, the packets which are sought to be forwarded, actions sought to be taken, ect., and based on the steps in FIG. 17, Sheet 17 of 20, a rules table entry for a rules table is then constructed (substantively the same as "programming the new entry does not involve storing the data packet" in the instant invention). It would have been obvious to one of ordinary skill in the art to combine the invention of Ambe with the invention of Park since Ambe provides a filter that can be programmed for use in routing packets and where the rules and information programmed in are stored in a rules table entry, saving the rules and information for future use in treating and forwarding certain packets or packet types, expanding the routing capabilities of Park.

In the same field of endeavor, Kadambi teaches in lines 23-24, 29-30, 44-47, column 27, of a device that receives a packet and "If the packet is identified as neither

an IP packet nor an IPX packet, the packet is directed to CPU", but the device could otherwise process a received IP packet or IPX packet since it can provide "support of both IP and IPX protocol", where this "capability is provided within logic circuitry". Kadambi teaches in lines 35-38, column 27, that this identification of packet type is done before cache lookup since packet type influences lookups (substantively the same as "prior to determining whether the multiple-key decision is hit" and "determining whether hardware circuitry of the network device is capable of processing the data packet" and "processing the data packet using hardware circuitry if the hardware circuitry is determined to be capable" in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt a form of the method of Park into the architecture of Kadambi since Kadambi provides a detailed architecture where at least aspects of Parks method can be implemented since both endeavor to improve the rate of forwarding though the use of caches and hashing.

In the same field of endeavor, Zhang teaches in column 6, lines 7-17, and in column 7, lines 6-13 and 18-26, network address translation between a dynamic IP address space and a permanent IP address space using stored network address translation (NAT) information, and where NAT information comprises a cache of mobility information generally including a plurality of cache entries, and where NAT information is stored and managed in a distributed cache (substantively the same as "a cached action comprises network address translation of a data packet" in the instant invention). It would have been obvious to one of ordinary skill in the art to combine the invention of Zhang with the invention of Park since Zhang provides a system of a cache that

contains network address translation information, which can be incorporated into the system of Park to allow the cache of Part to accommodate NAT so that the system of Park can support NAT when it is implemented as a node between disparate networks that have different addressing types or where one network is implementing private addressing.

Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over Park in view of Navada, Musoll, Ambe, Zhang, and further in view of Spinney (Patent Number: 5,414,704), hereafter referred to as Spinney.

As for Claim 4, as discussed above with respect to claim 1, Park teaches a method. Park fails to teach using multiple fields of a data packet and a hash value relating to multiple fields in the data packet is used in programming the new entry. Navada teaches using multiple fields of a data packet, and Spinney teaches a hash value relating to multiple fields in the data packet is used in programming the new entry.

In the same field of endeavor, Navada teaches in paragraphs [0028] and [0002], and Fig. 2, Sheet 2 of 8, "the key 216 may be a hardware address, software address, and/or VLAN tag", and "keys are typically Internet protocol (IP) addresses, media access control (MAC) addresses, virtual local area network (VLAN) tags, and other network identifiers" (substantively the same as "using multiple fields of a data packet" in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt the method of Navada into the method of Park since the method

of Park does not explicitly involve fast VLAN lookups, while the method of Navada explicitly provides a detailed method of memory efficient fast VLAN lookups.

In the same field of endeavor, Spinney teaches in lines 23-26, column 16, a packet's "hashed address 87 is used to index into the hash table 89 to select a hash bucket 90, and this selected hash bucket contains a pointer 93 into the translation table 94" (substantively the same as "a hash value relating to multiple fields in the data packet is used in programming the new entry" in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt the method of Spinney into the method of Park since Spinney provides extensive details of efficiently searching a large database where a hashing function provides many collisions when dealing with large numbers of entries.

Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over Park in view of Navada, Musoll, Ambe, Zhang, and further in view of Lawler et al. (Patent Number: 5,978,951), hereafter referred to as Lawler.

As for Claim 6, as discussed above with respect to claim 1, Park teaches a method. Park does not teach of an exclusive-or operation on packet IP addresses to generate a hash value. Lawler teaches these limitations. In the same field of endeavor, Lawler teaches in lines 60-62, column 14, of a device that "takes the sixteen bit CRCs generated on the SA and DA and XORs them together to generate a conversation based hash", where SA is Source Address, DA is Destination Address (see Lawler, lines 61-62, column 3). (Substantively the same as "the hash value is generated by

applying an exclusive-or operation to a source IP address and a destination IP address in the data packet" in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt the management unit of Lawler into the method of Park since Lawler provides an age table and detailed cache manipulations to provide flexible management that may be implemented to enhance the method of Park.

Claims 11, 13 and 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Park in view of Navada, Musoll, Ambe, Zhang, and further in view Murthy et al. (Patent Number: 5,610,905), hereafter referred to as Murthy.

As for Claims 11 and 13, as discussed above with respect to claim 1, Park teaches a method. Park does not teach of the clearing and possible population of a cache due to modification of a table. Murthy teaches these limitations. In the same field of endeavor, Murthy teaches in lines 51-53, column 14, and FIG. 12, Sheet 13 of 22, of a "Forwarding Table 80. This data structure is a two-dimensional array. One index of the array is RPORT 85", and, in lines 34-37, column 16, if "the correspondence between SA 16 and RPORT 85 is found to be invalid, all Bridging Cache entries 79 with the corresponding SA 16 value in the RPORT sub-cache 77 must be cleared (the "flush" step in FIG. 16)", with the intent that the cache will be refilled with valid entries as the network device continues operation using the new RPORT value, and in lines 42-44, column 16, "the offending Bridging Cache entries 79 must be removed or, if it is more efficient, all cache entries may be invalidated" (substantively the same as "if a modification of a pertinent table is detected, then the decision cache is cleared and

populated if possible" in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt the method of Park into the apparatus of Murthy since Park provides an efficient usage of two caches (a main cache and an instant cache) that can improve upon the cache system of Murthy.

As for Claim 14, as discussed above with respect to claim 1, Park teaches a method. Park does not teach that if a forwarding table is modified, a decision cache is cleared. Murthy teaches these limitations.

In the same field of endeavor, Murthy teaches in lines 51-53, column 14, and FIG. 12, Sheet 13 of 22, of a "Forwarding Table 80. This data structure is a two-dimensional array. One index of the array is RPORT 85", and, in lines 34-37, column 16, if "the correspondence between SA 16 and RPORT 85 is found to be invalid, all Bridging Cache entries 79 with the corresponding SA 16 value in the RPORT sub-cache 77 must be cleared (the "flush" step in FIG. 16)" (substantively the same as "if a modification of a pertinent table is detected, then a corresponding entry in the decision cache is cleared" in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt the method of Park into the apparatus of Murthy since Park provides an efficient usage of two caches (a main cache and an instant cache) that can improve upon the cache system of Murthy.

Claim 12 rejected under 35 U.S.C. 103(a) as being unpatentable over Park in view of Navada, Musoll, Ambe, Zhang, Murthy, and further in view of Voit et al. (Patent No.: US 6,798,751 B1), hereafter referred to as Voit.

As for Claim 12, as discussed above with respect to claim 1, Park teaches a method. Park does not teach a NAT table, ACL, layer 3 forwarding table, or layer 2 forwarding table. Navada teaches these limitations. Navada teaches in paragraph [0022], "IP address, port address, and hardware address tables" (substantively the same as "a network layer 3 forwarding table, and a network layer 2 forwarding table" in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt the method of Navada into the method of Park since the method of Park does not explicitly involve fast VLAN lookups, while the method of Navada explicitly provides a detailed method of memory efficient fast VLAN lookups.

In the same field of endeavor, Voit teaches in lines 29-30, column 8, "access control lists", and in line 64, column 33, "network address translations" (substantively the same as "a network address translation (NAT) table, an access control list (ACL)" in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt the method of Park into the equipment of Voit since Voit does not explicitly include a cache table or a method in which to utilize it for faster processing and forwarding of packets.

Claim 15 is rejected under 35 U.S.C. 103(a) as being unpatentable over Park in view of Navada, Musoll, Ambe, Zhang, and further in view of Luijten et al. (Patent Number: 6,023,466), hereafter referred to as Luijten.

As for Claim 15, as discussed above with respect to claim 1, Park teaches a method. Park does not teach if a modification of a pertinent table is detected, a corresponding entry in the decision cache is updated. Murthy teaches if a modification of a pertinent table is detected, and Luijten teaches a corresponding entry in the decision cache is updated.

In the same field of endeavor, Murthy teaches in lines 51-53, column 14, and FIG. 12, Sheet 13 of 22, of a “Forwarding Table 80. This data structure is a two-dimensional array. One index of the array is RPORT 85”, and, in lines 34-37, column 16, if “the correspondence between SA 16 and RPORT 85 is found to be invalid, all Bridging Cache entries 79 with the corresponding SA 16 value in the RPORT sub-cache 77 must be cleared (the "flush" step in FIG. 16)”, with the intent that the cache will be refilled with valid entries as the network device continues operation using the new RPORT value, and in lines 42-44, column 16, “the offending Bridging Cache entries 79 must be removed or, if it is more efficient, all cache entries may be invalidated” (substantively the same as “if a modification of a pertinent table is detected” in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt the method of Park into the apparatus of Murthy since Park provides an efficient usage of two caches (a main cache and an instant cache) that can improve upon the cache system of Murthy.

In the same field of endeavor, Luijten teaches in lines 27-29, column 4, “the entries to the LOOKUP RAM have to be kept directly addressable by the addresses to the corresponding entries of the SEARCH RAM. As a consequence, every update of the SEARCH RAM is accompanied by a corresponding update of the LOOKUP RAM” (substantively the same as “a corresponding entry in the decision cache is updated” in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt the method of Luijten into the method of Park since Luijten provide detailed techniques in managing and searching two separate but cooperating memory units without the use of a hash.

Claim 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over Park in view of Navada, Musoll, Ambe, Kadambi, Zhang, and further in view of Spinney.

As for Claim 19, Park shows in paragraph [0060] and in FIG. 5, Sheet 3 of 3, “if it is determined from the step S10 that the destination address of the packet is identical to the cache address found in the step S9”, then “the main processor 11 sends the packet to the interface corresponding to the cache address” (substantively the same as “determining whether pertinent fields of the data packet exactly match corresponding field of the entry, and wherein if the pertinent fields exactly match, then providing a result that the decision cache is hit” in the instant invention). Park fails to teach a hash value relating to multiple fields in a data packet used in programming a new entry, and hashing these fields and utilizing the result in a new table entry. Navada teaches using

multiple fields of a data packet, and Spinney teaches a hash value relating to multiple fields in the data packet is used in programming the new entry.

In the same field of endeavor, Navada teaches in paragraphs [0028] and [0002], and Fig. 2, Sheet 2 of 8, "the key 216 may be a hardware address, software address, and/or VLAN tag", and "keys are typically Internet protocol (IP) addresses, media access control (MAC) addresses, virtual local area network (VLAN) tags, and other network identifiers" (substantively the same as "using multiple fields of a data packet" in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt the method of Navada into the method of Park since the method of Park does not explicitly involve fast VLAN lookups, while the method of Navada explicitly provides a detailed method of memory efficient fast VLAN lookups.

In the same field of endeavor, Spinney teaches in lines 23-26, column 16, a packet's "hashed address 87 is used to index into the hash table 89 to select a hash bucket 90, and this selected hash bucket contains a pointer 93 into the translation table 94" (substantively the same as "a hash value relating to multiple fields in the data packet is used in programming the new entry" in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt the method of Spinney into the method of Park since Spinney provides extensive details of efficiently searching a large database where a hashing function provides many collisions when dealing with large numbers of entries.

Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over Park in view of Navada, Musoll, Ambe, Kadambi, Zhang, Spinney, and further in view of Lawler.

As for Claim 20, as discussed above with respect to claim 18, Park teaches a method. Park does not teach of an exclusive-or operation on packet IP addresses to generate a hash value. Lawler teaches these limitations.

In the same field of endeavor, Lawler teaches in lines 60-62, column 14, of a device that “takes the sixteen bit CRCs generated on the SA and DA and XORs them together to generate a conversation based hash”, where SA is Source Address, DA is Destination Address (see Lawler, lines 61-62, column 3). (Substantively the same as “the hash value is generated by applying an exclusive-or operation to a source IP address and a destination IP address in the data packet” in the instant invention). It would have been obvious to one skilled in the art at the time of the invention to adopt the management unit of Lawler into the method of Park since Lawler provides an age table and detailed cache manipulations to provide flexible management that may be implemented to enhance the method of Park.

Response to Arguments

4. Applicant's arguments with respect to claims 1, 2 and 4-20 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to JOSHUA SMITH whose telephone number is (571)270-1826. The examiner can normally be reached on Monday-Thursday 9:30am-7pm, Alternating Fridays 9:30am-6pm, EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hassan Kizou can be reached on 571-272-3088. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Joshua Smith
Patent Examiner

/Hassan Kizou/
Supervisory Patent Examiner, Art Unit 2619